CLASSIFICATION OF TRAFFIC PATTERN

Edward Chung, Visiting Professor
Centre for Collaborative Research, University of Tokyo,
4-6-1 Komaba, Meguro-ku, Tokyo, JAPAN 153-8904
Tel: +81-3-5452 6098, Fax: +81-3-5452 6420, Email: edward@nishi.iis.u-tokyo.ac.jp

SUMMARY

The aim of this study was to determine the travel pattern along the inbound route 3 of the Tokyo Metropolitan Expressway with the purpose of applying the result to classify historical travel time database used for travel time prediction. A data mining cluster algorithm SLR was used for the cluster analysis. This analysis found that AM period could be classified into weekday, Saturday and holiday (including Sunday). The clustering result suggests that the PM period does not have any strong groups.

INTRODUCTION

Travel time is often displayed on variable message signs installed on freeways, in car navigation system such as VICS[1] (Vehicle Information and Communication Systems) and on the internet. There are over 11 million vehicles in Japan with car navigation systems and of these over 6.5 million are installed with VICS which provides real time travel information such as travel time and location of accidents. Sales of VICS units are rapidly growing (see Figure 1).

On route travel time information allow drivers to make better route choices if alternative routes are available, whilst pre-trip travel time information allow trip makers to choose their mode and also the time to travel. Hence travel time information has the potential to mitigate congestion spatially and temporally. Furthermore, travel information has been reported by many researchers to reduce the stress of driving and results in safer driving as drivers know what to expect ahead of them.

Therefore with the benefits of travel time information and the quick market penetration of VICS, there is a need for more accurate travel time prediction model.

A common method (instantaneous travel time) used to estimate current travel time is by summing the travel time derived from speed measurements at different sections of the road simultaneously. The instantaneous travel time calculation assumes that present traffic condition would prevail for vehicles entering the road section now. This assumption is valid in free flow condition but as congestion starts building up, the instantaneous travel time starts lagging. Needless to say, the instantaneous travel time method is not suitable for predicting travel time at a longer time horizon of say 1 hour.

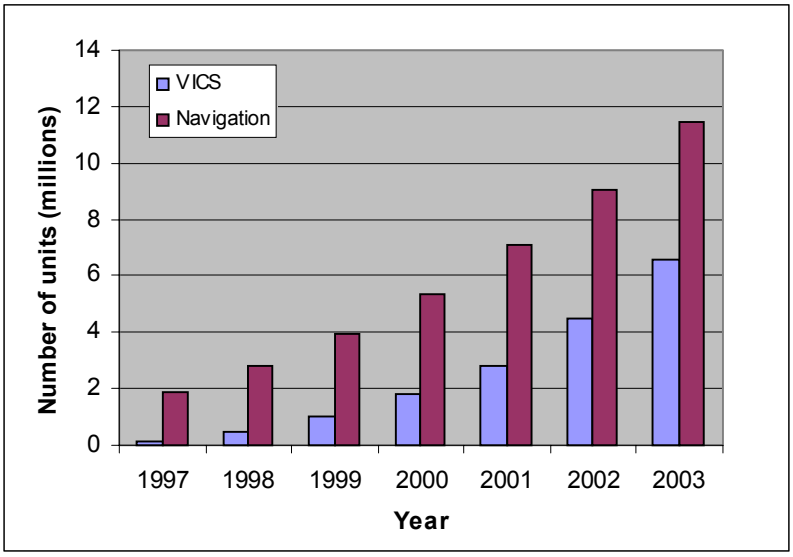

Figure 1: Market penetration of car navigation units in Japan

There are other methodologies proposed for predicting travel times ranging from very naïve statistical approaches to more sophisticated artificial intelligence and machine learning based algorithms. A simple and accurate technique is pattern matching of present traffic pattern with historical traffic pattern [2].

Pattern matching technique is based on the premise that traffic scenarios similar to present traffic condition have occurred before. A database of historical traffic scenarios is created for searching the closest *N* patterns (see Figure 2). The predicted short term travel time is taken as the average of the best matched historical patterns excluding outliers.

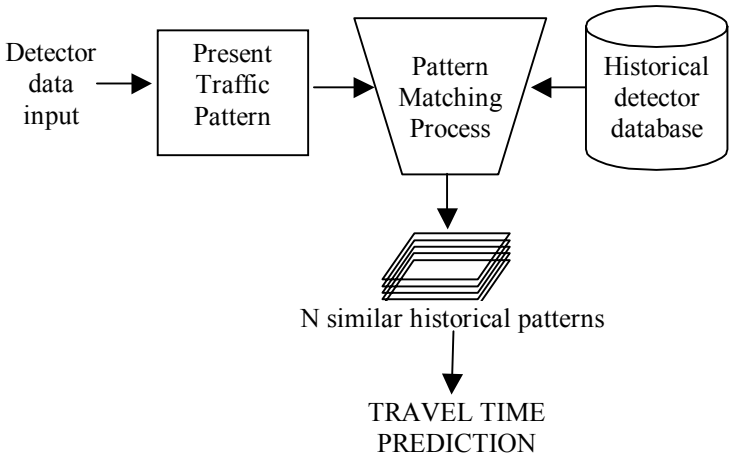

Figure 2: Travel time prediction model using pattern matching. Source: [3]

The computer time required for searching historical pattern can be excessive if the historical database is large (ie. contain many days of traffic pattern) or the search time window is large. Search time window is the time frame of $\pm x$ minutes of prediction time that traffic patterns of all days in the historical database are searched. The search time window is used, as it is unlikely that traffic situations will recur exactly at the same time as they occurred in the past. An effective way to reduce the computation time is to classify the historical database so that only similar segment of the historical database is searched. For example Sunday traffic pattern may be unique from the other days of the week, and if only Sunday historical traffic

patterns were searched for the prediction of Sunday travel time, the number of searches on a historical database with one year's traffic pattern would be reduced to 1/7 (52 Sundays per 365 days). Furthermore the number of outliers in the *N* best matched pattern may be reduced and better still the removal of outliers may not be necessary.

This paper discusses the classification of travel time on the Tokyo Metropolitan Expressway and the classification results. The following sections describe the sites where the travel time were measured, the clustering technique used and how the travel time data was structured to fit the clustering algorithm. Finally the clustering results and comparison of travel time prediction accuracy with classified and non-classified historical database are presented.

SITE DESCRIPTION

The travel time database used for this study was collected from the inbound direction of Route no. 3 of the Tokyo Metropolitan Expressway. Route 3 is a two lane expressway with a length of approximately 12km. It has three on ramps and three off ramps. The travel time on this route varies from 10 minutes in free flow condition to 90 minutes in severe congestion.

Ultrasonic detectors were installed approximately 300m apart to collect speed, flow and occupancy data. For this research, 2 years of detector data from April 2000 to March 2002 were used. Travel times were calculated every 5 minutes from the detector data by using the time slice method which estimates travel time by historical reconstruction of a vehicle's linear trajectory as the vehicle enters each section (halfway between 2 detector stations) using the speed measured from detectors installed on the route. The estimated travel time is sufficiently close to the actual travel time experienced by the vehicle because it uses the traffic condition of the section when the vehicle enters the section.

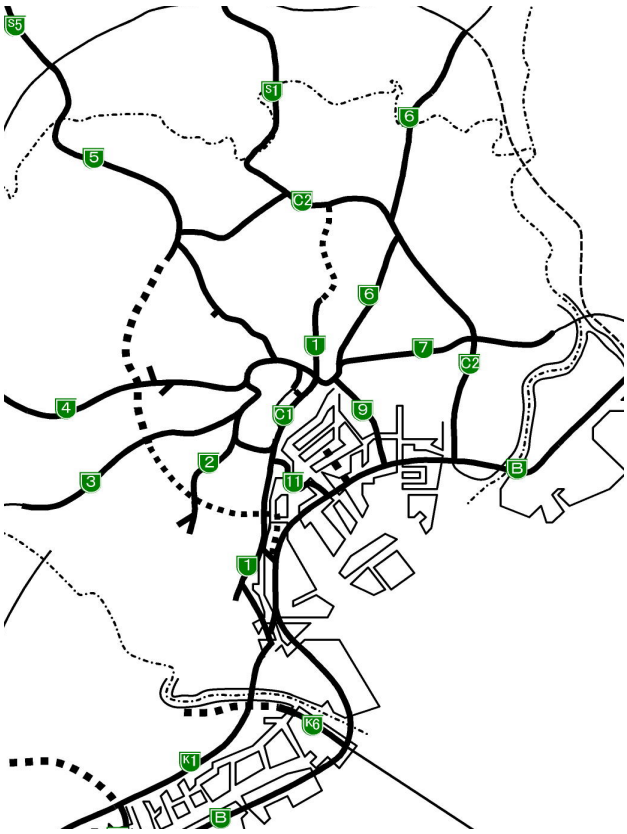

Figure 3 Tokyo Metropolitan Expressway Network

CLUSTERING ALGORITHM

There are 2 main types of clustering techniques, hierarchical and non hierarchical clustering. Hierarchical cluster analysis either begins with every data sample as a separate cluster, which are subsequently combined step by step until only one cluster remains (otherwise known as agglomerative hierarchical clustering), or as in the case of divisive hierarchical clustering, all entities are initially combined into one cluster and then separated into smaller clusters.

The non hierarchical cluster technique has two methods. The first separates the database into clusters in a single step by either maximising or minimising some numerical criteria (single pass method). The second method, called reallocation method, reallocates records from one cluster to another in order to create better clusters.

SMALL LARGE RATIO (SLR) ALGORITHM

The clustering technique used in this study is a reallocation non hierarchical method called *small large ratio (SLR) clustering algorithm for market basket data* [4]. Market basket data takes its name from items purchase in a shopping trolley. Market basket data analysis is the most common data mining techniques used to determine what products customers purchase together during grocery shopping (association rules). The SLR algorithm uses the notion of large and small items to perform the clustering. A large item is an item frequently found in a sufficient number of transactions and a small item is an item that appears in a limited number of transactions. Transaction is represented by a set of items purchased which is similar to the transaction of a customer's purchase at checkout counter. The support of an item *i* in a cluster *C* is defined as the percentage of transactions which contain this item *i* in cluster *C*. An item in a cluster is called a large item if the support of that item exceeds a pre-specified minimum support, *S*. Conversely, an item in a cluster is called a small item if the support of that item is less than a pre-specified maximum ceiling *E*. An item that is neither large nor small is called a middle item.

Using the example given in [4] and the support for items shown in Figure 4, if S=60% and E=30%, the large, middle and small items as shown in Figure 5 is obtained. In cluster C_3, D and H are large items whereas B, C and G are small items. The portions of large and small items represent the quality of the clustering as large items measure the similarity of a cluster and small items contribute to dissimilarity in a cluster. Yun et al. [4] developed an efficient clustering algorithm using the ratio of small items to large items in a cluster. Hence it is called the SLR algorithm and the smaller the SLR ratio, the more similar the items in that cluster are.

The SLR algorithm consists of two phases, the allocation phase and the refinement phase. The allocation phase assigns each transaction to an existing cluster or a new cluster to minimise the total cost. The cost function consists of intra-cluster cost and inter-cluster cost. The first measures the intra-cluster item dissimilarity and the second measure the inter-cluster item similarity. In other words, the cost is minimised if the intra-cluster items are similar and the inter-cluster items are dissimilar. The algorithm then moves all transactions whose SL ratio exceeds the pre-specified SLR threshold α (defined as excess transaction) into an excess pool, where all excess transactions are collected together. After collecting all excess transactions, the intermediate support values of items for identifying the large items and small items in each cluster are computed again. The algorithm then searches for the best cluster to reallocate all transactions in the excess pool. If there is no movement in an iteration after all transactions are scanned in the excess pool, the refinement phase terminates. An overview of the SLR algorithm is shown in Figure 6 and for further details, refers to [4].

C_1

TID	Items Set
110	B, D
120	A, B, D
130	B, C, D
140	D, F, H
150	B, G, I

C_2

TID	Items Set
210	B, I
220	A, B, I
230	B, E, I
240	B, C, E, I
250	C, I

C_3

TID	Items Set
310	D, H
320	D, F, H
330	B, C, D, F
340	H
350	D, G, H

Counting support

Item	Support
A	20%
B	80%
C	20%
D	80%
F	20%
G	20%
H	20%
I	20%

Item	Support
A	20%
B	80%
C	40%
E	40%
I	100%

Item	Support
B	20%
C	20%
D	80%
F	40%
G	20%
H	80%

Figure 4 Example database for clustering market basket data

Minimum support S=60%, Maximum ceiling E=30%

Cluster	Large	Middle	Small
C_1	B, D		A, C, F, G, H, I
C_2	B, I	C, E	A
C_3	D, H	F	B, C, G
	$Intra(U_0) = 7$ $Inter(U_0) = 2$ $Cost(U_0)=9$		

SLR Threshold = 3/2

C_1

TID	Items Set	SL Ratio
110	B, D	0/2
120	A, B, D	1/2
130	B, C, D	1/2
140	D, F, H	2/1
150	B, G, I	2/1

C_2

TID	Items Set	SL Ratio
210	B, I	0/2
220	A, B, I	1/2
230	B, E, I	0/2
240	B, C, E, I	0/2
250	C, I	0/1

C_3

TID	Items Set	SL Ratio
310	D, H	0/2
320	D, F, H	0/2
330	B, C, D, F	2/1
340	H	0/1
350	D, G, H	1/2

Figure 5 – Large, middle, small items in clusters and the corresponding SL ratios of transactions

Allocation phase: Assign each transaction t to an existing or a new cluster C_i to minimise Cost(U)

Refinement phase:

Step 1: calculate each cluster's minimum support, large items and small items;
Step 2: move all excess transactions from each cluster to excess pool;
Step3: again calculate each cluster's minimum support, large items and small items;
Step 4: for all excess transactions t_p in the excess pool, search for the best cluster C_j that t_p will have the smallest SLR in C_j; and move t_p to cluster C_j
Step 5: Iterate steps 1 to 4 until either no excess transactions can be moved or the excess pool is empty.

Figure 6 – Overview of SLR algorithm

RESEARCH APPROACH

The travel time database used in this study is a 5 minute travel time series of Route 3 inbound travel time. However the cluster algorithm SLR selected for this research requires input in the form of market basket data ie. each transaction with different items. The issue is how to structure the time series database so that they are suitable for the SLR algorithm. The following steps demonstrate the step used in this study.

Time period

The day is divided into 2 periods, AM and PM between 7 am to 1pm and 3pm to 8pm, respectively. The reason is that the AM pattern and PM pattern are very different for any particular day of the week. This is explained in the results section.

Smoothing of travel time

Haar wavelet is used to transform the travel time time-series to "smooth" the curve. As there are ripples in the 5 minute travel time series, wavelet helps to "smooth" out the ripples (see Figure 7). The Haar wavelet technique computes the means and differences of a signal [5][6]. Therefore, each step of the wavelet transform produces a set of means and a set of differences (also referred to as wavelet coefficients) that is half the size of the input data. For example, this study uses a time series of 64 elements, the first transformation produces 32 means and 32 coefficients. The means then become the input for the next transformation. This research uses only 2 steps resulting in a set of 16 means and 16 coefficients.

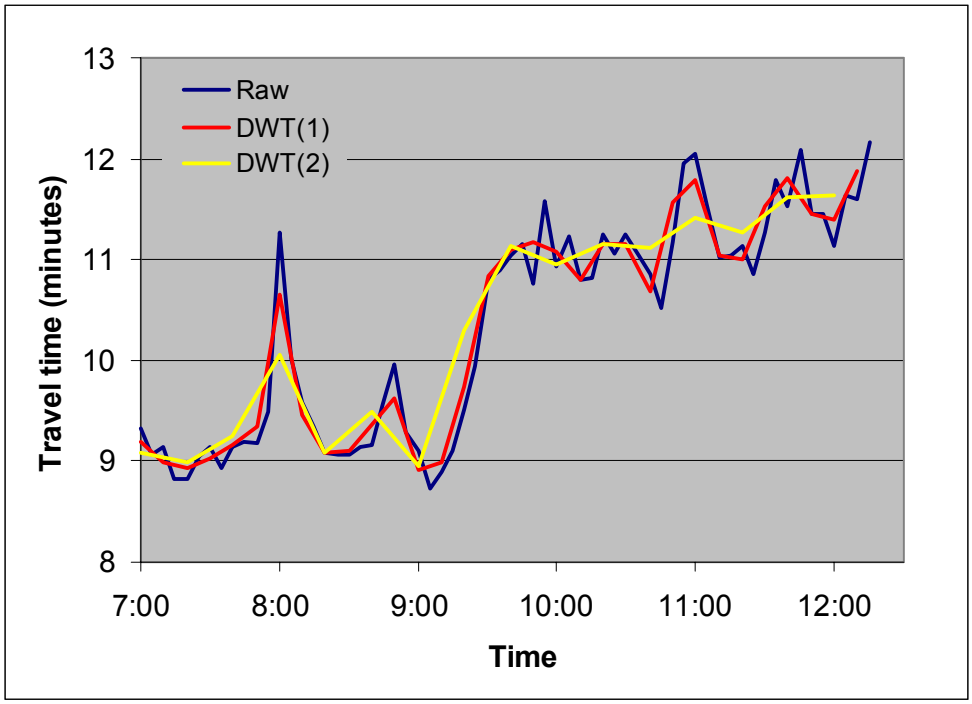

Figure 7 – Wavelet transformation of travel time

Fitness matrix

A fitness matrix of the squared difference between each day is calculated. This forms a 730 by 730 size matrix. The day closest matched to the subject day will have the smallest squared difference and vice versa. Using this relationship, the days closest matched to the subject day is ranked in ascending order.

Exogenous database

Information regarding the amount of rainfall (per day, maximum per hour, maximum per 10 minutes), time of sunrise and sunset and holidays were collated for the 2 year study period. Categories were introduced to each attribute so that each day can be assigned a code depending on the attributes of the day. The categories were:

- **Day of week** – 1 for Monday and 7 for Sunday
- **Rainfall** – 0 for 0-5 mm/day, 1 for >5 mm – 20mm/day and 2 for > 20 mm/day
- **Long weekend** –Code of 0, 1, 2 and 4 were used for not a long weekend, day after long weekend, long weekend and day before long weekend respectively.
- **Holiday** – A 5 digit binary code was used to represent 2 days before and 2 days after the subject day with the 3rd digit representing subject day (see Figure 8). Holiday is coded as 1 and non holiday as 0. The numeric value of the binary code which could range from 00 to 31 is then used. Note that the numerical value is a 2 digit field. For example if today is a holiday it would be represented as 00100 and the numeric value is 04 and if yesterday was a holiday and tomorrow will also be a holiday, the code is 01010 and has the value of 10.

Each day's attributes are used to create a code. For example, 5th April 2000 is a Wednesday (3) with heavy rainfall (2), not a long weekend (0) and not a holiday (00). This day will be coded as 32000.

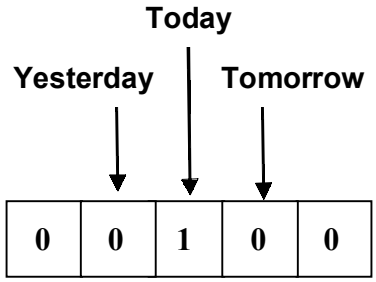

Figure 8 – Binary code for holiday

Market basket data

Combining the ranked fitness matrix and the day's attribute, the matrix can be transformed into a market basket data. Take 2nd April 2000 as an example, the 6 best matched days are shown in Table 1. Based on the attributes of each of these days and using the coding system explained above, a code can be assigned. However, the array of codes showed that some codes such as 70000 and 71000 are repeated. The next step is to only store codes that are unique for this group of days. Therefore by using the steps explained, a travel time time-series can be transformed into a group of unique code (ie. items in a transaction) which resembles a market basket data.

Table 1 Transformation of best matched days to market basket data format

Ranked days	**4/2/2000**	9/16/2001	3/31/2002	4/1/2001	8/15/2001	5/27/2001	4/9/2000
Ranked days in code	**70000**	70008	70000	70000	30000	71000	70000
Unique code	**70000**	70008	30000	71000			

CLUSTER ANALYSIS

Cluster analysis was undertaken for the 2 years of travel time data explained above. The objective was to find evidence of some inner structure for the data based on the similarities between the individual data. The SLR algorithm was applied to the transformed travel time data. First, a cluster analysis was undertaken for the AM travel time and another cluster analysis was undertaken for the PM travel time to investigate if there were any other distinguishing characteristics between the AM and PM traffic patterns.

There were a number of variables in the cluster analysis which could affect the outcome of the analysis. The variables are:

- **Closeness threshold** – this variable is used to determine which days are considered as good matches. Days with squared difference greater than the threshold value are ignored. Values used range from 50 to 1000.
- **Basket size** – this is the maximum number of items allowable in a basket. The basket size is also partially related to the closeness threshold as large closeness threshold would have more days in the group and hence more items in the basket. A sensible basket size is desirable. Values used range from 5 to 20.
- **Minimum support S** – S=60% was used.
- **Maximum ceiling E** –E=30% was used.
- **SLR threshold α** – Values used range from α =0.75 to α =3.0. Smaller α means that the items in each cluster will have greater similarity.

The final parameter values used for the cluster analysis were closeness threshold of 200, basket size of 10 and α =0.75 and 1.5 for AM and PM respectively. The outcome of the cluster analysis is discussed in the next section.

CLUSTERING RESULTS

Clustering is an unsupervised learning technique and as such there is no perfect correct answer. The AM cluster solution indicates that the weekday is a cluster with Saturday and Sunday as two separate clusters (see Table 2). The Sunday cluster also includes Monday holiday in this cluster suggesting that this is a *holiday* cluster (cluster 4 as shown in Table 2).

It is important to note that since the year 2000, Japan has moved most of its public holidays that fall on a weekday to either a Friday or Monday. This is a drive to make public holidays into long weekends so the public can enjoy themselves more and therefore spend more to boost the ailing Japanese economy. That is why no other weekdays were included in the *holiday* cluster.

Rainfall does not seem to be a factor in the clustering. In order to confirm this observation, the day attribute code was trimmed to remove the rainfall attribute and the SLR algorithm was rerun. The result was very similar to the result when full day attribute coding was used. Cluster 2 is a little different from the other clusters for the reason that it does not seem to have a common thread amongst the items in the cluster. This needs further investigation which will be addressed in the next section.

The PM cluster solution suggests Tuesday, Wednesday and Thursday as a group and Monday and Tuesday as another group. The remaining clusters consist of mostly individual day such as Friday, Wednesday or Thursday. Interestingly this cluster solution is consistent with field

observation. Since the travel time is for the inbound route, the evening travel time on a day where Monday is a holiday would see drivers returning to Tokyo.

In summary, the AM period could be grouped in weekday, Saturday and holiday (including Sunday). However, for the PM period, each day should be treated separately. Future research is required to determine whether to treat them as 7 individual groups or weekday, Saturday and Sunday.

Table 2 AM peak clusters

Cluster number	Number of days	Day attributes
Cluster 1	30	Saturday
Cluster 2	7	Tuesday after a long weekend, Wednesday where Tuesday was a holiday, Sunday where coming Tuesday is a holiday
Cluster 3	9	Thursday, Friday
Cluster 4	85	Monday holiday ie. long weekend, Sunday with no rain or light rain, Sunday where Friday or Saturday was a holiday
Cluster 5	227	Monday, Tuesday, Wednesday, Thursday, Friday
Cluster 6	18	Tuesday, Wednesday, Friday
Cluster 7	14	Monday, Monday with heavy rain
Cluster 8	23	Monday, Tuesday, Thursday

Table 3 PM peak clusters

Cluster number	Number of days	Day attributes
Cluster 1	22	Wednesday
Cluster 2	95	Monday, Tuesday
Cluster 3	18	Friday
Cluster 4	3	Tuesday where Sunday and Friday were holidays
Cluster 5	172	Tuesday, Wednesday, Thursday
Cluster 6	7	Friday with light rain
Cluster 7	43	Thursday, Sunday
Cluster 8	44	Friday
Cluster 9	21	Thursday

TRAVEL TIME PREDICTION WITH CLASSIFICATION

One of the benefits of a classified database is that the search time for pattern matching is reduced as the whole database is segmented. However, the accuracy of the travel time may be compromised as a result of the classification. A test was carried out for the period between 7am to 2pm. Using the AM cluster solution above, the historical travel time database was classified into weekday, Saturday and holiday. The travel time prediction model developed

by Bajwa [2] was modified to search for pattern from the newly classified historical database (see Figure 9). Three Mondays that were public holidays were tested and the results are shown in Table 4.
The comparison shows that the classified historical database performed worse than the non-classified database in only one out of the three days. This result is encouraging and more thorough clustering analysis and comparative study should be carried in the future.

Table 4 Comparison of travel time prediction accuracies

Date	Historical database	Correl. Coeff.	Mean Absolute Error	Mean Absolute % Error	Percentage of prediction	
					within 5%	within 10%
24 Dec 2001	classified	0.829	0.4	4.4	70.2	89.3
14 Jan 2002	classified	0.784	0.3	3.4	84.5	92.9
11 Feb 2002	classified	0.942	0.7	5.1	69.0	86.9
24 Dec 2001	non- classified	0.877	0.4	4.0	75.0	90.5
14 Jan 2002	non- classified	0.768	0.3	3.5	84.5	92.9
11 Feb 2002	non- classified	0.923	0.8	5.5	69.0	84.5

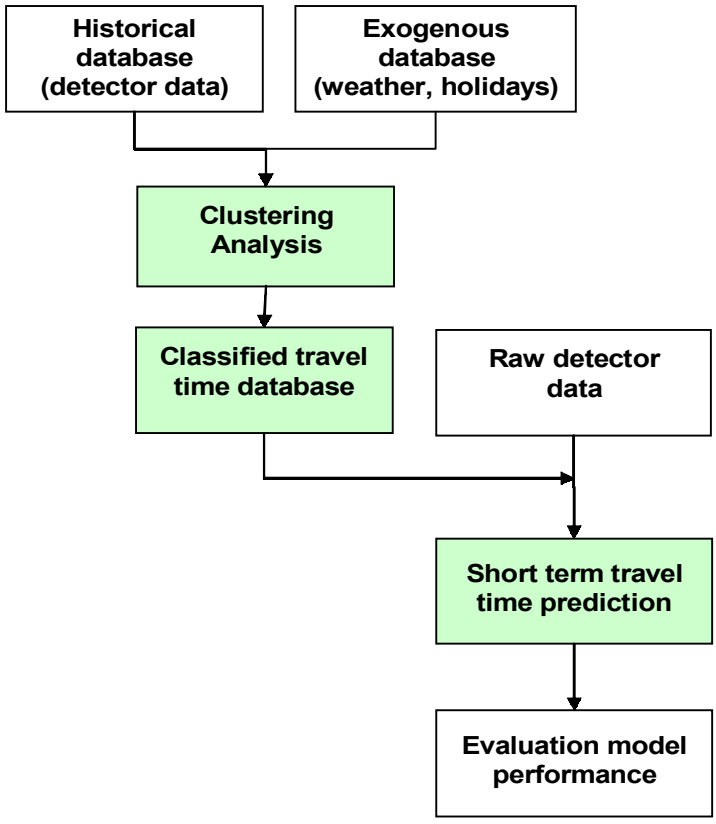

Figure 9: Travel time prediction model using classified database

CONCLUSION

The aim of this study was to determine the travel pattern along the inbound route 3 of the Tokyo Metropolitan Expressway with the purpose of applying the result to classify historical travel time database used for travel time prediction. A data mining cluster algorithm SLR was

used for the cluster analysis. The research approach developed in this study was able to transform the travel time time-series into market basket data format. This analysis found that AM period could be classified into weekday, Saturday and holiday (including Sunday). The clustering result suggests that the PM period does not have any strong groups. The AM classification was put to the test and found that the overall accuracy of the travel time prediction model did not deteriorate after classifying the historical database. In fact, 2 out of 3 test days were performing better than the traditional (non-classified method).

This study only looked at three attributes ie. day of the week, rainfall and holiday. Further research should be carried out to include more attributes such as the maximum travel time, shape of the travel time profile and month of the year.

ACKNOWLEDGEMENTS

The author would like to thank Tokyo Metropolitan Expressway Public Corporation for making the data available for this research, H. Warita for compiling the exogenous database and S. Bajwa for running his travel time prediction model. The support of other members of the Travel Time Research Group at Kuwahara Laboratory is also gratefully acknowledged.

REFERENCES

[1] VICS - Vehicle Information and Communication System. http://www.vics.or.jp

[2] S. Bajwa, E. Chung and Masao Kuwahara (2003). "A travel time prediction method based on pattern matching technique". *21st ARRB and 11th REAAA Conference*, CD, 18 - 23 May, 2003

[3] S. Bajwa, E. Chung and Masao Kuwahara (2003). "Sensitivity analysis of short-term travel time prediction model's parameters". *ITS World Congress, Madrid, Nov*, 2003

[4] C. Yun, K. Chuang and M. Chen (2001). "An efficient clustering algorithm for market basket data based on small large ratios". *25th Annual International Computer Software and Applications Conference (COMPSAC'01)*, pp. 505-510, 8-12 October, 2001

[5] A. Jensen and A. la Cour-Harbo (2001) "Ripples in Mathematics", *Springer Verlag.*

[6] I. Kaplan (2002) "Wavelets and Signal Processing"
http://www.bearcave.com/misl/misl_tech/wavelets/index.html